Grad-CAM:
Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju[1*] Michael Cogswell[1] Abhishek Das[1] Ramakrishna Vedantam[1*]
Devi Parikh[1,2] Dhruv Batra[1,2]
[1]Georgia Institute of Technology [2]Facebook AI Research
{ramprs, cogswell, abhshkdz, vrama, parikh, dbatra}@gatech.edu

Contents

1. Class Activation Map (B. Zhou et al., 2015.)
2. Guided Backpropagation (J. T. Springenberg et al., 2015.)
3. Grad-CAM

Class Activation Map (B. Zhou et al., 2015.)

- Global Average Pooling (M. Lin et al., 2014.)
- class에 해당하는 weight들로 마지막 conv layer에 weighted sum
- 해당 class가 활성화되는 부분을 확인!

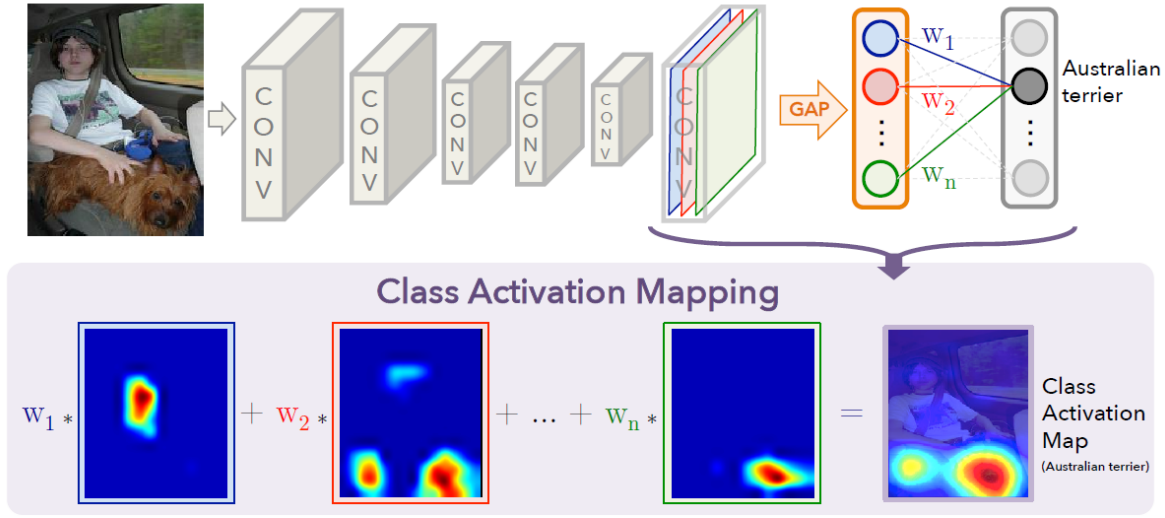


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y)$$

$$= \sum_{x,y} \sum_k w_k^c f_k(x,y). \quad (1)$$

$$M_c(x,y) = \sum_k w_k^c f_k(x,y). \quad (2)$$

Guided Backpropagation (J. T. Springenberg et al., 2015.)

- 기존 backpropagation은 forward의 activation 방식을 따름
- deconvnet (M. D. Zeiler et al., 2014.) 은 gradient의 값으로 activation을 적용
- Guided Backpropagation은 backpropagation + deconvnet

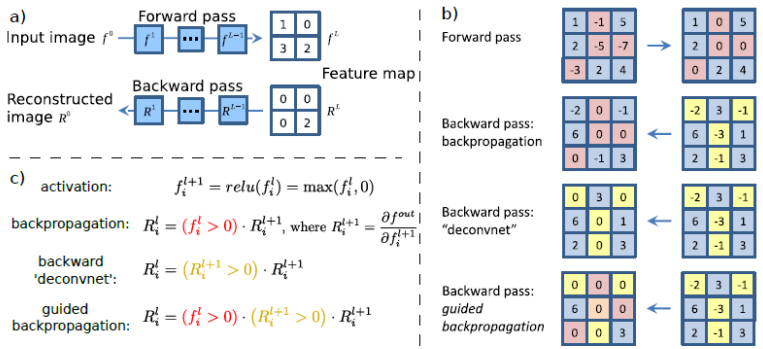


Figure 1: Schematic of visualizing the activations of high layer neurons. a) Given an input image, we perform the forward pass to the layer we are interested in, then set to zero all activations except one and propagate back to the image to get a reconstruction. b) Different methods of propagating back through a ReLU nonlinearity. c) Formal definition of different methods for propagating a output activation *out* back through a ReLU unit in layer l; note that the 'deconvnet' approach and guided backpropagation do not compute a true gradient but rather an imputed version.

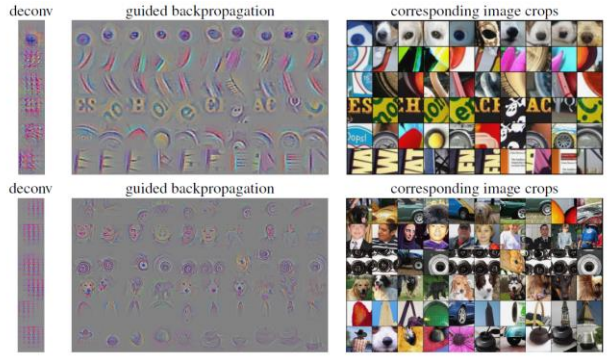


Figure 3: Visualization of patterns learned by the layer conv6 (top) and layer conv9 (bottom) of the network trained on ImageNet. Each row corresponds to one filter. The visualization using "guided backpropagation" is based on the top 10 image patches activating this filter taken from the ImageNet dataset. Note that image sizes are not preserved (in order to save space).

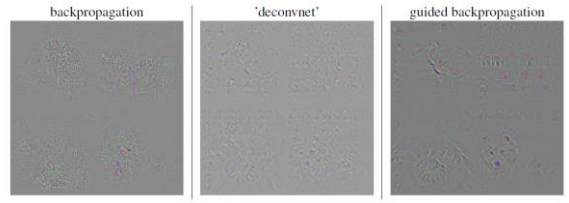


Figure 6: Visualization of descriptive image regions with different methods from the single largest activation in the last layer fc8 of the Caffenet reference network (Jia et al., 2014) trained on ImageNet. Reconstructions for 4 different images are shown.

Grad-CAM

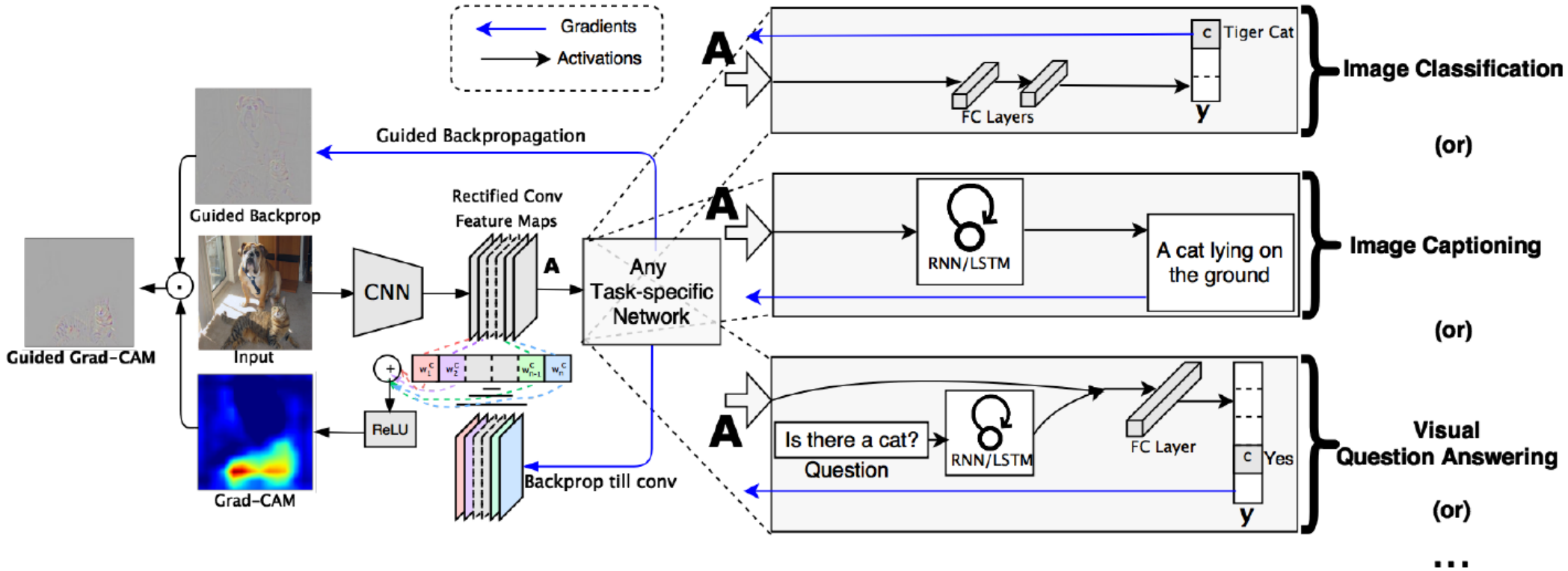


Figure 2: Grad-CAM overview: Given an image and a class of interest (*e.g.*, 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

Grad-CAM

- CAM은 GAP(Global Average Pooling)을 사용해야만 하고, 재학습이 필수적임!

  → gradient를 계산해서 그것으로 weighted sum

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \tag{1}$$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}} \tag{2}$$

$$S^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}} \tag{3}$$

$$S^c = \frac{1}{Z}\sum_i\sum_j\underbrace{\sum_k w_k^c A_{ij}^k}_{L^c_{\text{CAM}}} \tag{4}$$

Grad-CAM

- CAM은 GAP(Global Average Pooling)을 사용해야만 하고, 재학습이 필수적임!

  → layer가 깊을수록 뚜렷한 효과가 나타남

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \qquad (1)$$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}} \qquad (2)$$

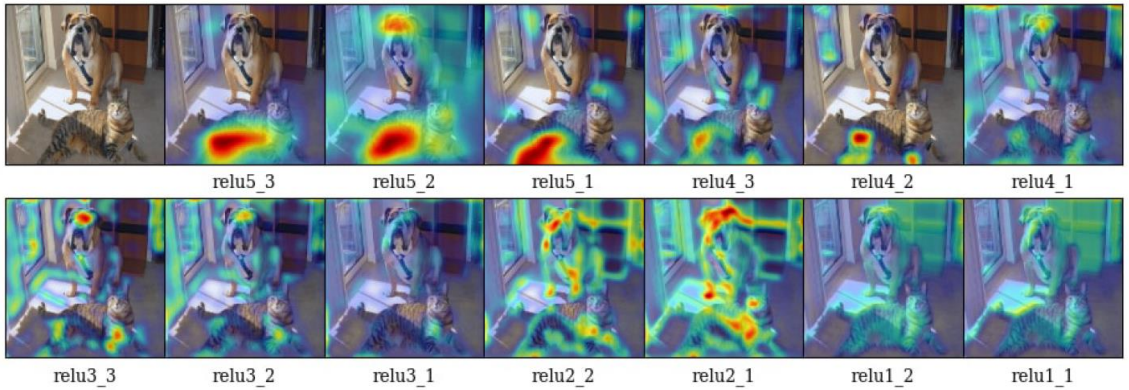


Figure A6: Grad-CAM at different convolutional layers for the 'tiger cat' class. This figure analyzes how localizations change qualitatively as we perform Grad-CAM with respect to different feature maps in a CNN (VGG16 [45]). We find that the best looking visualizations are often obtained after the deepest convolutional layer in the network, and localizations get progressively worse at shallower layers. This is consistent with our intuition described in Section 3 of main paper.

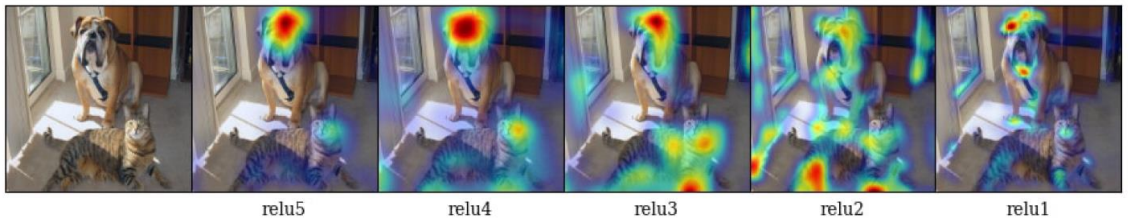


Figure A7: Grad-CAM localizations for "tiger cat" category for different rectified convolutional layer feature maps for AlexNet.

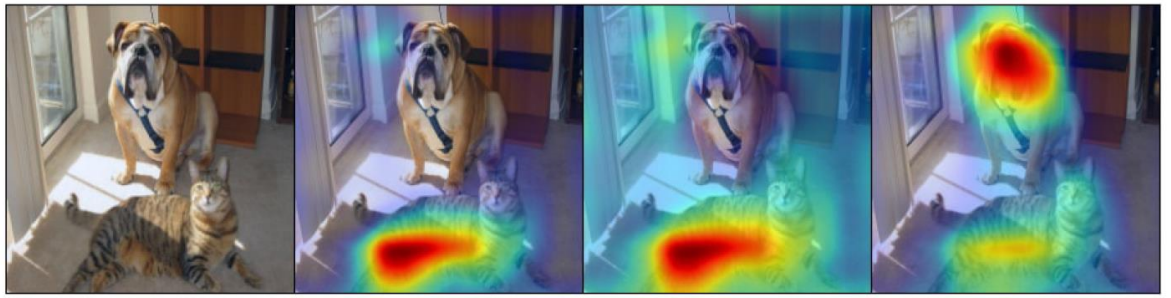
Grad-CAM

- CAM은 GAP(Global Average Pooling)을 사용해야만 하고, 재학습이 필수적임!

  → gradient에 ReLU를 적용해서 positive값만 사용

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \quad (1)$$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}} \quad (2)$$



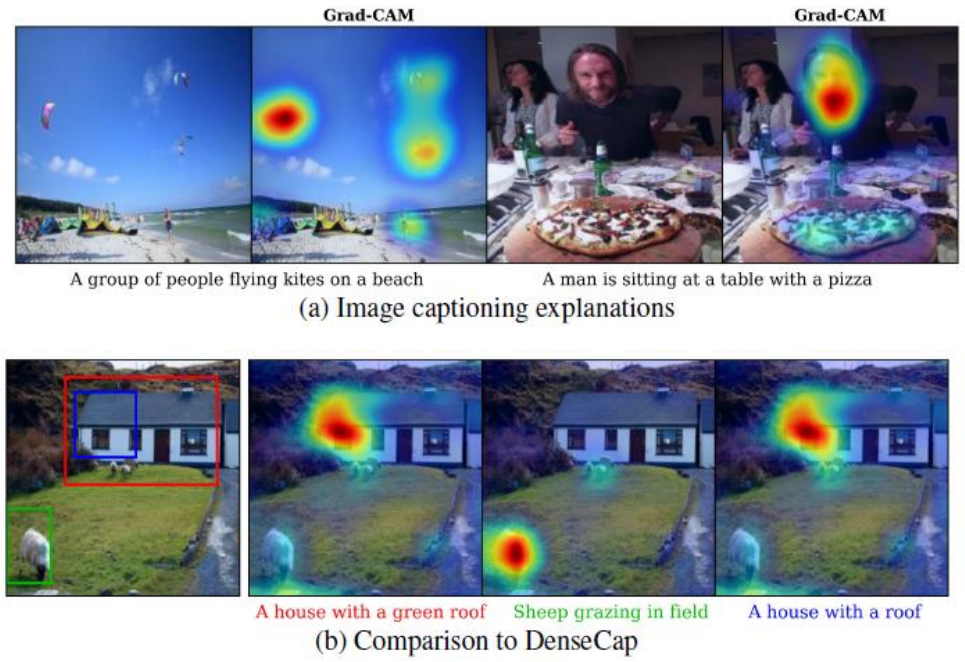
Grad-CAM | Grad-CAM without ReLU | Grad-CAM with absolute gradients

Figure A8: Grad-CAM visualizations for "tiger cat" category stating the importance of ReLU and effect of using absolute gradients in Eq. 1 of main paper.

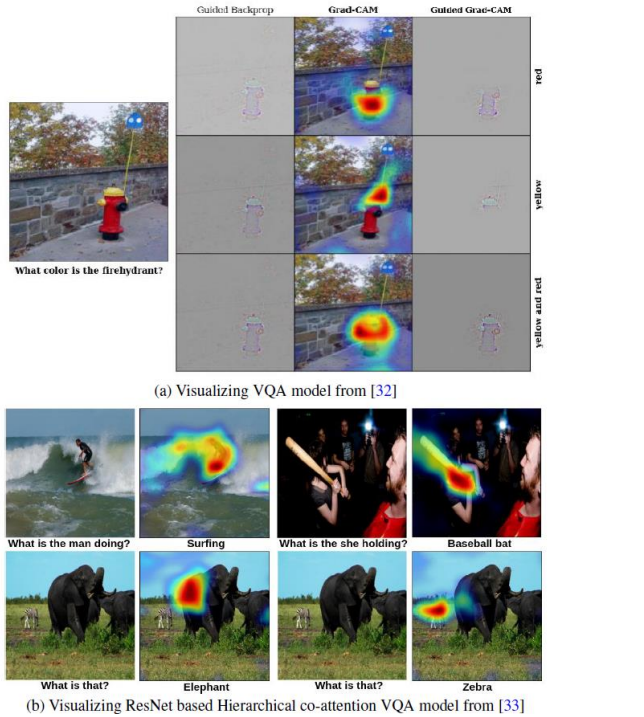
Grad-CAM

- CAM은 GAP(Global Average Pooling)을 사용해야만 하고, 재학습이 필수적임!

  → 다양한 task에서 사용 가능



Captioning



Visual Question Answering (VQA)

Grad-CAM

- Guided Backpropagation과 결합하여 사용 가능 (Guided Grad-CAM)

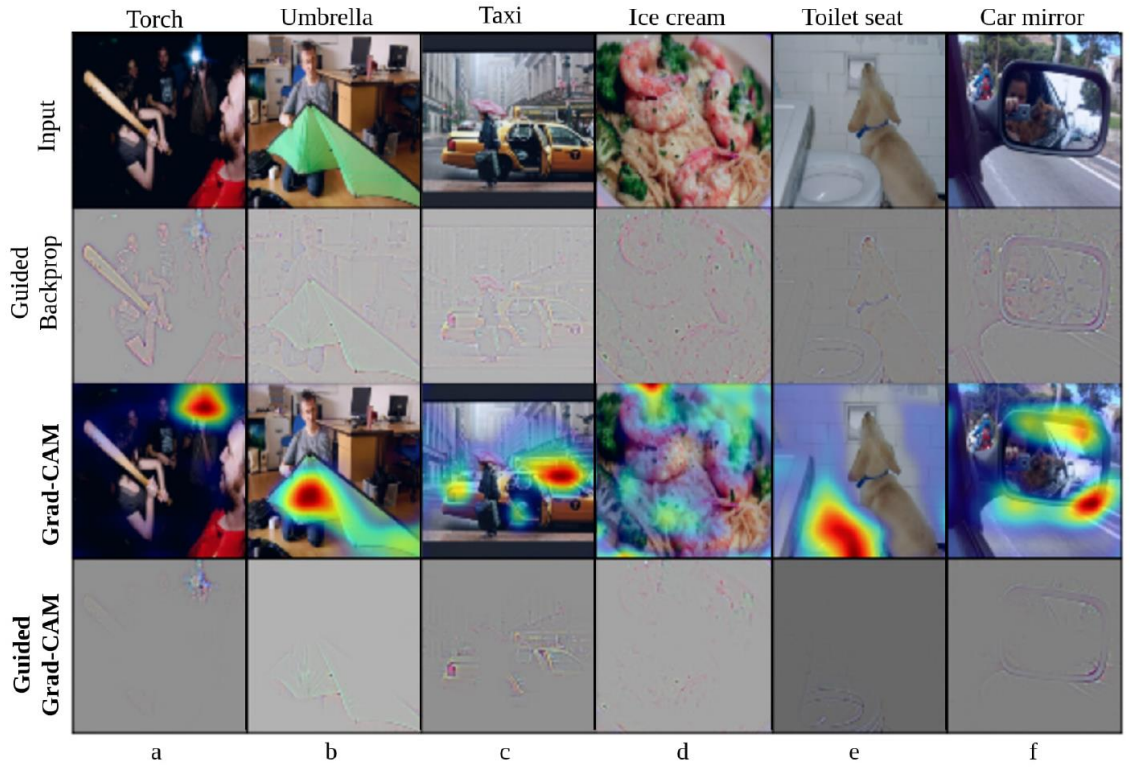


Figure A1: Visualizations for randomly sampled images from the COCO validation dataset. Predicted classes are mentioned at the top of each column.

Grad-CAM

- Guided Backpropagation과 결합하여 사용 가능 (Guided Grad-CAM)

  → 마찬가지로 다양한 task에 적용 가능

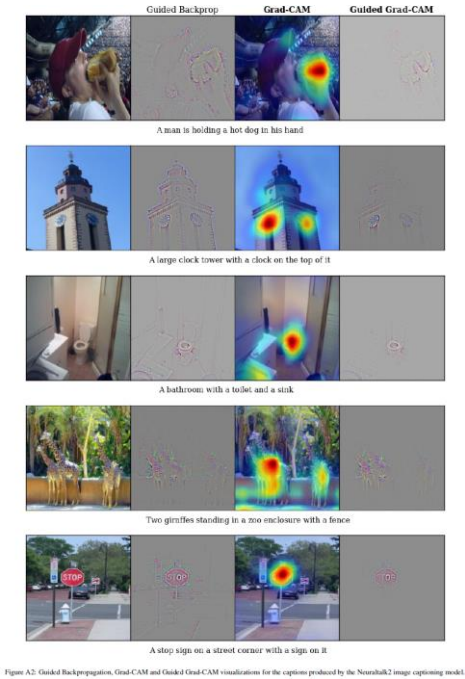


Figure A2: Guided Backpropagation, Grad-CAM and Guided Grad-CAM visualizations for the captions produced by the Neuraltalk2 image captioning model.

Captioning

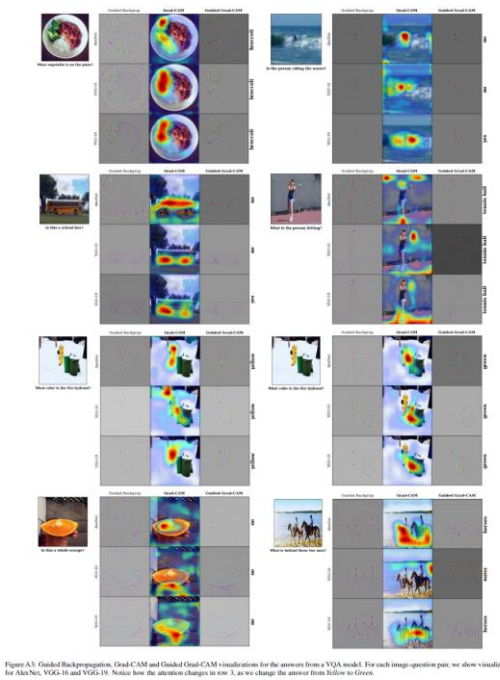


Figure A3: Guided Backpropagation, Grad-CAM and Guided Grad-CAM visualizations for the answers from a VQA model. For each image-question pair, we show visualizations for AlexNet, VGG-16 and VGG-19. Notice how the attention changes in row 3, as we change the answer from *Yellow* to *Green*.

Visual Question Answering (VQA)

Grad-CAM

- Guided Backpropagation과 결합하여 사용 가능 (Guided Grad-CAM)

  → 제대로 예측하지 못하는 부분을 알아내어 문제를 해결할 수 있음!

Figure A4: Grad-CAM explanations for model1 and model2. In (a-c) we can see that even though both models made the right decision, the biased model (model1) was looking at the face of the person to decide if the person was a nurse (b), whereas the unbiased model, was looking at the short sleeves to make the decision (c). For example image (d) and example (g) the biased model made the wrong prediction (misclassifying a doctor as a nurse) by looking at the face and the hairstyle (e, h), where as the unbiased model made the right prediction looking at the white coat, and the stethoscope (f, i).

Reference

- R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, *"Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization"*, in ICCV, 2017.
- R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh and D. Batra, *"Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization"*, in CORR, abs/1610.02391, 2016.
- B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, *"Learning Deep Features for Discriminative Localization"*, in CVPR, 2016.
- M. Lin, Q. Chen, and S. Yan. *"Network in network"*, in ICLR, 2014.
- J. T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, *"Striving for Simplicity: The All Convolutional Net"*, in ICLR, 2015.
- M. D. Zeiler and R. Fergus, *"Visualizing and understanding convolutional networks"*, in ECCV, 2014.